

Document Generated: 02/20/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 5 Days

Next Course Date: **April 6, 2026**

Programming in C# - Creating Applications in C# and .Net Core (TTCN20483)



About This Course:

Geared for experienced developers, Programming in C# / Creating Apps in C# and .Net Core (20483) is a five day, hands-on course that provides you with the skills

required to create applications in C# and .Net Core. Throughout the hands-on course you'll explore and learn the basics of Visual C# program structure, language syntax, and implementation details, and then consolidate their knowledge throughout the week as they build an application that incorporates several features of the .NET Core 5. This independent course aligns with the topics and skills in Microsoft Official Curriculum (MOC) course 20483.

Course Objectives:

- Describe the core syntax and features of Visual C#.
- Create methods, handle exceptions, and describe the monitoring requirements of large-scale applications.
- Implement the basic structure and essential elements of a typical desktop application.
- Create classes, define and implement interfaces, and create and use generic collections.
- Use inheritance to create a class hierarchy and to extend a .NET Framework class.
- Read and write data by using file input/output and streams, and serialize and deserialize data in different formats.
- Create and use an entity data model for accessing a database and use LINQ to query data.
- Access and query remote data by using the types in the System.Net namespace and WCF Data Services.
- Build a graphical user interface by using XAML.
- Improve the throughput and response time of applications by using tasks and asynchronous operations.
- Integrate unmanaged libraries and dynamic components into a Visual C# application.
- Examine the metadata of types by using reflection, create and use custom attributes, generate code at runtime, and manage assembly versions.
- Encrypt and decrypt data by using symmetric and asymmetric encryption.

Audience:

- Developers attending this course should already have gained some limited experience using C# to complete basic programming tasks

Prerequisites:

More specifically, students should have hands-on experience using C# that demonstrates their understanding of the following:

- How to name, declare, initialize and assign values to variables within an application.
- How to use: arithmetic operators to perform arithmetic calculations involving one or more variables; relational operators to test the relationship between two variables or expressions; logical operators to combine expressions that contain relational operators.
- How to create the code syntax for simple programming statements using C# language keywords and recognize syntax errors using the Visual Studio IDE.
- How to create a simple branching structure using an IF statement.
- How to create a simple looping structure using a For statement to iterate through a data array.
- How to use the Visual Studio IDE to locate simple logic errors.
- How to create a Function that accepts arguments (parameters and returns a value of a specified type).
- How to design and build a simple user interface using standard controls from the Visual Studio toolbox.
- How to connect to a SQL Server database and the basics of how to retrieve and store data.
- How to sort data in a loop.
- How to recognize the classes and methods used in a program.

Course Outline:

Chapter 1: Review of C# Syntax & New Features

Get started writing clean, modern C# code with updated features and the tools you will use throughout the course.

- Understand Visual Studio and project setup
- Compare .NET Core and .NET Framework
- Create simple console applications
- Use top-level statements effectively

- Apply global usings to clean code
- Explore core C# building blocks

Chapter 2: Creating Methods, Handling Exceptions, and Monitoring Applications

Learn how to organize your code, manage errors gracefully, and keep applications running smoothly using real-world techniques.

- Write and call reusable methods
- Create overloaded methods and optional parameters
- Handle exceptions to improve stability
- Monitor applications using logs and tracing
- Track errors with real-world techniques
- Build smarter, more reliable applications

Chapter 3: Basic Types and Constructs of C#

Build a solid foundation by working with core data types, collections, and events to structure your application logic.

- Create and use structs and enums
- Organize data with collections
- Store and retrieve items easily
- Loop through data the right way
- Subscribe to and handle events
- Make code flexible with core types

Chapter 4: Creating Classes and Implementing Type-Safe Collections

Learn how to design custom types that are easy to reuse, maintain, and apply to real projects.

- Define and create classes
- Build and use interfaces
- Organize code for reusability
- Implement generic collections safely
- Make types easy to maintain
- Structure applications the smart way

Chapter 5: Creating a Class Hierarchy by Using Inheritance

Use inheritance to build connected class structures that let you reuse logic and extend your applications easily.

- Create base and derived classes
- Extend functionality through inheritance
- Reuse code across related types
- Simplify changes with structured hierarchy
- Improve application flexibility with design
- Practice real-world class structures

Chapter 6: Reading and Writing Local Data

Work with local files and data streams to store, read, and manage information efficiently within your applications.

- Read and write files with C#
- Serialize and deserialize application data
- Use streams for large data handling
- Have and load data easily
- Work with different file formats
- Connect applications to local storage

Chapter 7: Accessing a Database

Learn how to connect to a database, retrieve and update data, and build useful data-driven features.

- Set up Entity Framework connections
- Query databases using LINQ
- Manage data through C# code
- Build data-driven application features
- Perform basic create, read, update, delete
- Explore real-world database scenarios

Chapter 8: Accessing Remote Data

Send and receive information across the web to build more connected and useful applications.

- Send requests to web services
- Receive and process remote data
- Connect applications to external APIs
- Work with online data sources
- Build lightweight web-connected apps
- Handle remote data safely

Chapter 9: Designing the User Interface for a Graphical Application

Explore different ways to build user interfaces for desktop or web apps that look good and respond to users.

- Create MAUI apps for cross-platform use
- Build simple UIs with Blazor
- Develop ASP.NET web applications
- Understand UI layout basics
- Connect user actions to functionality
- Make user-friendly and responsive designs

Chapter 10: Improving Application Performance and Responsiveness

Keep your apps fast and responsive by learning techniques that prevent delays and manage background work.

- Use tasks for multitasking operations
- Handle long-running processes smoothly
- Apply `async` and `await` correctly
- Coordinate work across multiple threads
- Improve app speed and responsiveness
- Build better user experiences

Chapter 11: Creating Reusable Types and Assemblies

Add flexibility to your applications by creating reusable code and working with advanced features like attributes and reflection.

- Inspect code with reflection
- Create and use custom attributes
- Work with object metadata
- Package reusable code cleanly
- Add flexible functionality to apps
- Build smarter, scalable applications